УДК 517.958

ПАРАЛЛЕЛЬНЫЙ ПРЕДОБУСЛАВЛИВАТЕЛЬ AISM ДЛЯ РЕШЕНИЯ СЛАУ НА ГРАФИЧЕСКИХ УСКОРИТЕЛЯХ[1)]

Н.С. НЕДОЖОГИН, С.П. КОПЫСОВ

Институт Механики УрО РАН
E-mail Nedozhogin@inbox.ru

PARALLEL PRECONDITIONER AISM FOR SOLVING SYSTEMS ON THE GPUs

N.S. NEDOZHOGIN, S.P. KOPYSOV

Institute of Mechanics Ural Branch of Russian Academy of Sciences

Аннотация

Исследуются возможности ускорения предобусловленных итерационных методов бисопряженных градиентов (BiCGStab) с предобуславливателем на основе разложения матрицы по формуле Шермана-Моррисона. Рассмотрена эффективность распараллеливания наиболее ресурсоёмких операций данного предобуславливателя на многоядерных центральных процессорах и графических процессорах.

Ключевые слова: решение СЛАУ, предобуславливание, итерационный метод, параллельные вычисления, формула Шермана-Моррисона

Summary

The capability of acceleration of the preconditioned iterative methods of biconjugate gradient method with preconditioner based on the decomposition on the matrix by the Sherman-Morrison formula are investigated. The efficiency of parallelization of the most resource-intensive operations in this preconditioner on the multi-core CPUs and GPUs are considered.

Key words: Solving systems of the linear algebraic equations, preconditioning, iterative method, parallel computing, Sherman-Morrison formula.

Введение

Построение эффективных методов решения больших систем линейных алгебраических уравнений на основе предобусловленных итерационных методов, особенно в контексте параллельных вычислений, является достаточно трудной задачей. Матрица предобуславливатель не только должна быть в определенном смысле близка к обратной матрице коэффициентов системы, но и должна допускать эффективно распараллеливаемый алгоритм ее формирования и умножения на вектор.

К широко используемым сегодня можно отнести методы, ориентированные на разреженные матрицы и основанные на неполном разложении на треугольные составляющие, такие как метод неполного LU-разложения и метод неполного разложения Холецкого [1]. Имея высокую эффективность и популярность данные алгоритмы сталкиваются с известными проблемами при их параллельной реализации.

Высокий потенциал распараллеливания имеют предобуславливатели на основе аппроксимации обратной матрицы: полиномиальные, разреженные аппроксимации обратной матрицы (например, AINV), аппроксимации обратной матрицы в факторизованной форме (такие как FSAI, SPAI и др.) [2], а также метод AISM, основанный на разложении матрицы по формуле Шермана-Моррисона (Sherman-Morrison formula) [3].

В работе ставятся следующие задачи: разработка методов параллельного построения предобуславливателей AISM; реализация и практическая оценка параллельной эффективности предложенных методов для гибридных вычислительных систем.

[1)] Работа выполнена при поддержке РФФИ (проекты 14-01-31066-мол-а, 14-01-00055-а)

1. Предобуславливатель AISM

Рассмотрим предобуславливатель, основанный на обращении матриц методом Шермана-Моррисона. Параллельные свойства алгоритма обращения оказались достаточно хорошими, как и ожидалось из теоретических построений [4]. В настоящей работе исследуется техника предобусловливания задачи, предложенная в [5], схема которой изложена далее.

За основу построения A^{-1} возьмем матрицу B той же размерности, что и A, но с известной обратной матрицей.

Теорема [3]. Пусть B – невырожденная матрица и вектора u и v такие, что $r = 1 + v^T B^{-1} u \neq 0$, матрица $A = B + uv^T$ является обратимой и её обращение находится как

$$A^{-1} = B^{-1} - r^{-1} B^{-1} u v^T B^{-1}. \tag{1}$$

Обозначим u_k, v_k и A_0 – невырожденная матрица, обращение которой просто вычисляется (т.е. A_0 может быть диагональной или даже единичной матрицей). Тогда $A_k = A_0 + \sum_{i=1}^{k} u_i v_i^T$, где $k = 1, n$ и $A = A_n$. Если A_k, u_k, v_k удовлетворяют выражению (1), тогда обращение матрицы A может быть вычислено при n-кратном использованием (1)

$$A^{-1} = A_0^{-1} - \sum_{k=1}^{n} r_k^{-1} A_{k-1}^{-1} u_k v_k^T A_{k-1}^{-1}. \tag{2}$$

Представим (2) в матричной форме

$$A_0^{-1} - A^{-1} = \Phi \Omega^{-1} \Psi^T, \tag{3}$$

где $\Phi = [A_0^{-1} u_1, A_1^{-1} u_2, \ldots, A_{n-1}^{-1} u_n]$, $\Psi = [v_1^T A_0^{-1}, u_2^T A_1^{-1}, \ldots, v_n^T A_{n-1}^{-1}]$ и $\Omega^{-1} = \mathrm{diag}\,[r_1^{-1}, r_2^{-1}, \ldots, r_n^{-1}]$. Покажем, что факторизация (3) записывается без явного вычисления A_k^{-1} через вектора u_k, v_k как

$$s_k = u_k - \sum_{i=0}^{k-1} \frac{t_i^T A_0^{-1} u_k}{r_i} s_i, \quad t_k = v_k - \sum_{i=0}^{k-1} \frac{v_k^T A_0^{-1} s_i}{r_i} t_i, \quad k = 1, \ldots, n. \tag{4}$$

Тогда выполняются соотношения

$$A_{k-1}^{-1} u_k = A_0^{-1} s_k, \quad u_k^T A_{k-1}^{-1} = t_k^T A_0^{-1}, \tag{5}$$

$$r_k = 1 + v_k^T A_0^{-1} s_k = 1 + t_k^T A_0^{-1} u_k. \tag{6}$$

С учетом (5) соотношение (3) запишем в виде

$$A_0^{-1} - A^{-1} = A_0^{-1} S \Omega^{-1} T^T A_0^{-1}, \tag{7}$$

где матрицы $S = [s_1, s_2, \ldots, s_n]$, $T = [t_1, t_2, \ldots, t_n]$ столбцы которых вычисляются по u_k, v_k.

Выбор A_0, u_k, v_k следующим образом [5]

$$A_0 = gI_n, \quad u_k = e_k, \quad v_k = (a^k - a_o^k)^T, \quad k = 1, \ldots, n.$$

где I_n, e_k – единичная матрица и ее k-й столбец, векторы a^k, a_0^k – k-я строки матриц A, A_0. Тогда аппроксимация обратной матрицы и соответствующий предобуславливатель примут вид

$$P_1 = A^{-1} = gI_n - g^{-2} U \Omega^{-1} V^T.$$

Рассмотрим вариант когда матрица представима в виде разложения $A = W - Z$, где W – обратимая матрица, $Z = UV^T = \sum_{k=1}^{n} u_k v_k^T$, а v_k, u_k такие, что $d_k = 1 - v_k^T W_{k-1}^{-1} u_k \neq 0$, где $W_k = W_0 - \sum_{i=1}^{k} u_u v_i^T$. Задавая выбор матриц $W = \beta \cdot diag(A)$, $\beta > 0$, $U = I$, $V = Z^T$ и следуя соотношениям (4), (5) и (7), получим выражения для вычисления столбцов матриц S и T

$$s_k = u_k - \sum_{i=1}^{k-1} \frac{t_i^T W^{-1} u_k}{d_i} s_i, \quad t_k = v_k - \sum_{i=1}^{k-1} \frac{v_k^T W^{-1} s_i}{d_i} t_i$$

и обратной матрицы в виде

$$A^{-1} = W^{-1} - W^{-1}SD^{-1}T^TW^{-1}. \quad (8)$$

Используя стратегию фильтрации по значениям элементов (оставляем элементы значения которых больше некоторой величины τ) при вычислении матриц S, T и основываясь на (8) выпишем предобуславливатель, аппроксимирующий обратную матрицу в виде

$$P = W^{-1} - W^{-1}\tilde{S}D^{-1}\tilde{T}^TW^{-1}. \quad (9)$$

Алгоритм построения рассматриваемого предобуславливателя представлен ниже.

Algorithm 1 Построение предобуславливателя AISM

1: $A = W - Z$
2: $W = \beta \cdot diag(A); Z = W - A$
3: $U = I; V = Z^T$
4: **for** $k = 1$ **to** n **do**
5: $s_k = u_k$
6: $t_k = v_k$
7: **for** $i = 1$ **to** k **do**
8: $temp = (t_i^T W^{-1}, u_k)$
9: **if** $|\frac{temp}{d_i}| > \tau_u$ **then**
10: $s_k = u_k - \frac{temp}{d_i} \cdot s_i$
11: **end if**
12: $temp = (v_k^T W^{-1}, s_i)$
13: **if** $|\frac{temp}{d_i}| > \tau_v$ **then**
14: $t_k = v_k - \frac{temp}{d_i} \cdot s_i$
15: **end if**
16: **end for**
17: **if** $|s_k| < \tau_u$ **then**
18: $s_k = 0$
19: **end if**
20: **if** $|t_k| < \tau_v$ **then**
21: $t_k = 0$
22: **end if**
23: $d_k = 1 - (t_k^T W^{-1}, u_k)$
24: **end for**
25: $\tilde{S} = \{s_1, s_2, \ldots, s_n\}, \tilde{T} = \{t_1, t_2, \ldots, t_n\}$
26: $D = \{d_1, d_2, \ldots, d_n\}$
27: $P = W^{-1} - W^{-1}\tilde{S}D^{-1}\tilde{T}^TW^{-1}$

2. Параллельное построение предобуславливателя.

Рассмотрим процесс построения предобуславливателя в модели параллелизма по данным, которая присуща вычислению скалярных произведений векторов.

При использовании технологии CUDA, для вычислений на графических ускорителях скалярные произведения векторов выполнялись с помощью функций `cublasDdot` (для случая с двойной точностью) и `cublsSdot` (с одинарной точностью). Остальные векторные операции были реализованы с помощью ядер (kernel) CUDA собственной разработки.

В рамках одной итерации цикла Алгоритма 1, при вычислении векторов s_k и t_k, не возникает ситуации блокировки памяти, что позволяет выполнять операции матрично-векторного и скалярных произведений (шаги 8, 12) независимо в параллельных нитях OpenMP. Однако, такой подход требует больших затрат при реализации для multiGPU варианта, так как каждая последующая итерация цикла зависит от данных, полученных на предыдущем шаге, и требуется производить обмен s_k и t_k между памятью различных GPU.

Кроме этого, для сравнения все операции над векторами (инициализация, умножения вектора на скаляр, сложение векторов) были реализованы также в модели общей памяти OpenMP с помощью `#pragma omp for`.

Последний шаг Алгоритма 1, содержащий матричные операции (умножение, сложение) был так же распараллелен как в рамках технологии CUDA, так и OpenMP. При использовании CUDA, было реализовано ядро, представляющее матрично-матричное произведение в виде последовательных матрично-векторных произведений.

Табл. 1: Число итераций при решении предобусловленных систем уравнений

Матрица	Cond(A)	$P_{diag(A)}$	P_{AISM} CPU	P_{AISM} GPU (FP32)	P_{AISM} GPU (FP64)
nasa2910	$9.53 \cdot 10^{64}$	1039	516	513	512
bcsstk15	$6.64 \cdot 10^{9}$	166	123	100	95
$\tau = 1 \cdot 10^{-6}$				71	
Kuu	$15.75 \cdot 10^{3}$	263	100	98	97
msc10848	$9.97 \cdot 10^{7}$	1693	980	1017	815
$\tau = 0.5$				1671	
$\tau = 1 \cdot 10^{-4}$				715	
vibrobox					
$\tau = 1 \cdot 10^{-4}$	$1.04 \cdot 10^{19}$	121	58	53	
FIDAP/ex19	$2.15 \cdot 10^{12}$	404	131	130	

При построении предобуславливателя P разреженность матрицы неизвестна. Промежуточные вычисления на этапе формирования выполнялись над векторами s_k, t_k, являющиеся столбцами матриц S, T, хранящихся по строкам. Расходы по памяти увеличивались, но сокращалось время обращения к элементам векторов. Для преобразования из сжатого формата хранения матриц CSR в формат хранения полных строк (этап построения предобуславливателя) и обратное преобразование (матрично-векторное произведение при решении СЛАУ) были разработаны эффективные параллельные алгоритмы, позволяющие пренебречь затратами на преобразование матриц.

3. Результаты численных экспериментов.

Часть тестовых задач выполнялась в пакете OpenFoam, в который была успешно выполнена интеграция программной реализации на GPU решения СЛАУ итерационным методом BiCGStab с параллельным предобуславливателем AISM.

Кроме того, в численных экспериментах были использованы матрицы из коллекции The University of Florida Sparse Matrix Collection. Решались системы уравнений с известным точным решением $[1, 1, \ldots, 1]$, матрицы которых хранились в сжатом строчном формате (CSR). В качестве начального приближения выбиралось $y_0 = [0, 0, \ldots, 0]$, а критерий сходимости – $||r_i|| \leqslant 10^{-6}||r_0||$, где $r_i = f - Ay_i$. Численные эксперименты проводились на GPU-ускорителе GeForce GTX 580, 3 ГБ и на восьми ядрах CPU (два четырехядерных процессора Intel Xeon E5430, 2.66 ГГц, 8 ГБ).

В численных экспериментах точность фильтрации для матриц S и T выбиралась $\tau_u = \tau_v = 0.01$, $\beta = 100$. Сравнение эффективности предобуславливателя P_{AISM} и диагонального масштабирования представлены в таблице 1.

Здесь же для матриц bcsstk15, msc10848 также приведены результаты с другими параметрами фильтрации $\tau = \tau_u = \tau_v$. Отметим, что в этом случае число итераций итерационного процесса существенно изменяется при сохранении общих вычислительных затрат формирования матрицы предобуславливателя P.

Оценка использование параллельных вычислений с одинарной точностью (FP32) на графическом ускорителе GeForce GTX 580 при построении предобуславливателя AISM показала, что влияние одинарной точности вычислений P сводится к незначительному увеличению числа итераций метода BiCGStab. Существенного прироста производительности не достигается, хотя теоретическая производительность данного GPU на операциях с одинарной точностью восьмикратно превышает вычисления с двойной точностью (FP64).

Затраты по памяти при хранении матриц S, T и P составляют $18 * n^2$ байт для варианта с двойной точностью и $12 * n^2$ – с одинарной, что накладывает ограничение на максимальный размер рассматри-

Табл. 2: Ускорение при формировании предобуславливателя

Матрица	$A(n/nnz)$	P_{AISM} OpenMP	P_{AISM} GPU(FP32)	P_{AISM} GPU (FP64)
nasa2910	2910 / 174296	1.67	1.49	1.01
bcsstk15	3948 / 117816	1.81	1.67	1.14
Kuu	7102 / 340200	1.61	1.87	1.38
msc10848	10848 / 1229778	1.88	3.23	2.57
vibrobox	12328 / 301700	1.68	2.61	2. 01
FIDAP/ex19	12005 / 259577	1.74	2.64	2. 06

ваемых систем для решения на GPU ($n\tilde{1}3000$). Хотя возможны варианты сокращения затрат памяти при хранении матриц S и T в одном из разреженных форматов.

Результаты сравнения параллельного и последовательного алгоритмов формирования предобуславливателя P_{AISM} для тестовых матриц сведены в табл. 2. Наибольшее ускорение достигнуто для матрицы msc10848 (3.23). Данная матрица имеет большую заполненность, в отличие от остальных, и параллельные матричные произведения при построении предобуславливателя P существенно ускоряются. Для остальных матриц достигается двукратное ускорение в случае OpenMP, и приближаются к трехкратному (в случае CUDA) с увеличением порядка системы. Вычислительная нагрузка операций скалярных произведений менее интенсивная, что связано с ограничением пропускной способности памяти и получить существенные ускорения возможно при больших порядках матриц и при использовании нескольких GPU.

Алгоритм AISM обладает определенными свойствами при распараллеливании, например, вычисления скалярных произведений. Время вычисления предобуславливателя остается значительным и превышает затраты итерационных приближений при решении СЛАУ, в отличии от предобуславливателя AINV, где затраты во многих случаях соизмеримы. Дальнейшее повышение параллельной эффективности предобуславливателя AISM должно быть связано с тем или иным блочным представлением алгоритма и выделением крупноблочной декомпозиции матрицы. Один из возможных вариантов – блочное представление для обратной матрицы в виде дополнения Шура [6]. В этом случае рассмотренный вариант параллельного предобуславливателя будет соответствовать второму уровню параллельной декомпозиции алгоритма.

ЛИТЕРАТУРА

1. **Saad Y.** Iterative Methods for Sparse Linear Systems. – SIAM, 2003.
2. **Benzi M.** Preconditioning Techniques for Large Linear Systems: A Survey // Journal of Computational Physics. – 2002. – V. 182, № 2. – P. 418–477.
3. **Sherman J., Morrison W.J.** Adjustment of an inverse matrix corresponding to a change in one element of a given matrix // Ann. Math. Statistics. – 1950. – V. 21, № 1. – P. 124–127.
4. **Недожогин Н.С., Сармакеева А.С., Копысов С.П.** Высокопроизводительный алгоритм Шермана-Моррисона обращения матриц на GPU // Вестник ЮУрГУ. Серия «Вычислительная математика и информатика». – 2014. – Т. 3, № 2. – С. 101–108.
5. **Bru M., Cerdán J., Marín J., Mas J.** Preconditioning sparse nonsymmetric linear systems with Sherman-Morrison formula // SIAM J. Sci. Comput. – 2003. – V. 28, № 2. – P. 701 –715.
6. **Копысов С.П., Кузьмин И.М., Недожогин Н.С., Новиков А.К.** Параллельные алгоритмы формирования и решения системы дополнения Шура на графических ускорителях // Ученые записки Казанского университета. Серия Физико-математические науки. – 2012. – Т. 154, кн. 3. – С. 202–215.

REFERENCES

1. **Saad Y.** Iterative Methods for Sparse Linear Systems. – SIAM, 2003.
2. **Benzi M.** Preconditioning Techniques for Large Linear Systems: A Survey // Journal of Computational Physics. – 2002. – V. 182, № 2. – P. 418–477.
3. **Sherman J., Morrison W.J.** Adjustment of an inverse matrix corresponding to a change in one element of a given matrix // Ann. Math. Statistics. – 1950. – V. 21. № 1. – P. 124–127.
4. **Nedozhogin N.S., Sarmakeeva A.S., Kopysov S.P.** High-performance Sherman-Morrison algorithm of the matrix inversion on the GPU [Vysokoproizvoditel'nyy algoritm Shermana-Morrisona obrashcheniya matrits na GPU] // Vestnik YUUrGU. Vychislitel'naya matematika i informatika. – 2014. – V. 3, № 2. – P. 101–108 (in Russian).
5. **Bru M., Cerdán J., Marín J., Mas J.** Preconditioning sparse nonsymmetric linear systems with Sherman-Morrison formula // SIAM J. Sci. Comput. – 2003. – V. 28, № 2. – P. 701 –715.
6. **Kopysov S.P, Kuzmin I.M., Nedozhogin N.S., Novikov A.K.** Parallel algorithms for forming and solving the Schur complement system on graphics accelerators [Parallel'nyye algoritmy formirovaniya i resheniya sistemy dopolneniya Shura na graficheskikh uskoritelyakh] // Uchenyye zapiski Kazanskogo universiteta. «Fiziko-matematicheskiye nauki». – 2012. – V. 154, № 3. – P. 202–215 (in Russian).